CLAIMS

We claim:

1. A method of monitoring processor resources, said method comprising:

determining a set of needed resources for a block of code;

testing if said set of resources are available at a start of said block of code; and

signaling an error if said set of resources needed for said block of code are not available.

2. The method as claimed in claim 1, said method further comprising:

determining a set of available resources that will be available after said block of code has executed.

3. The method as claimed in claim 1 wherein said needed resources comprise stack contents.

4. The method as claimed in claim 1 wherein said set of needed resources is determined at a compile time.

1   5.   The method as claimed in claim 1 wherein said set of needed

2   resources is determined dynamically.

1   6.   The method as claimed in claim 1 wherein signaling said

2   error if said set of resources needed for said block of code are not available

3   comprises branching to a fault handler routine.

1   7.   The method as claimed in claim 6 wherein signaling said

2   fault handler routine simulates a processor exception.

1   8.   The method as claimed in claim 1 wherein needed resources

2   are represented by a bit vector.

1   9.   The method as claimed in claim 8 wherein said bit vector is

2   generated dynamically.

1   10.   A computer-readable medium having stored thereon a set of

2   instructions to monitor processor resources, said set of instruction, which when

3   executed by a processor, cause said processor to perform a method comprising:

4   determining a set of needed resources for a block of code;

5   testing if said set of resources are available at a start of said block of code;

6   and

7   signaling an error if said set of resources needed for said block of code are

8   not available.

1   11.   The computer-readable medium as claimed in claim 10,

2   wherein said set of instructions further includes additional instructions, which

3   when executed by said processor, cause said processor to perform said method

4   further comprising:

5   determining a set of available resources that will be available after said

6   block of code has executed.

1   12.   The computer-readable medium as claimed in claim 10

2   wherein said needed resources comprise stack contents .

1   13.   The computer-readable medium as claimed in claim 10

2   wherein said set of needed resources is determined at a compile time.

14. The computer-readable medium as claimed in claim 10
2 wherein said set of needed resources is determined dynamically.

1 15. The computer-readable medium as claimed in claim 10
2 wherein signaling said error if said set of resources needed for said block of code
3 are not available comprises branching to a fault handler routine.

1 16. The computer-readable medium as claimed in claim 15
2 wherein signaling said fault handler routine simulates a processor exception.

1 17. The computer-readable medium as claimed in claim 10
2 wherein needed resources are represented by a bit vector.

1 18. The computer-readable medium as claimed in claim 17 wherein
2 said bit vector is generated dynamically.

19.   A computer-readable medium, having stored thereon a first

2   set of instructions, the first set of instructions, which when executed by a

3   processor, generate a second set of instructions through a binary translation

4   process, the second set of instructions when executed by the processor, cause

5   said processor to perform a method comprising:

6   determining a set of needed resources for a block of code;

7   testing if said set of resources are available at a start of said block of code;

8   and

9   signaling an error if said set of resources needed for said block of code are

10   not available.

1   20.   The computer-readable medium as claimed in claim 19,

2   wherein said set of instructions further includes additional instructions, which

3   when executed by said processor, cause said processor to perform said method

4   further comprising:

5   determining a set of available resources that will be available after said

6   block of code has executed.

1   21.   The computer-readable medium as claimed in claim 19

2   wherein said needed resources comprise stack contents .

22. The computer-readable medium as claimed in claim 19 wherein said set of needed resources is determined dynamically.

23. The computer-readable medium as claimed in claim 19 wherein signaling said error if said set of resources needed for said block of code are not available comprises branching to a fault handler routine.

24. The computer-readable medium as claimed in claim 23 wherein signaling said fault handler routine simulates a processor exception.

25. The computer-readable medium as claimed in claim 19 wherein needed resources are represented by a bit vector.